Causal Modeling and lhcnt-driven Simulation for Monitoring of Continuous Systems

Richard Doyle, Leonard Charest, Jr., Nicolas Rouquette and Jay Wyatt
Jet Propulsion Laboratory
Pasadena, CA 91109-8099
Charles Robertson
McDonnell-Douglas Aerospace
Houston, TX 77059-3556

Abstract

Monitoring of complex continuous physical systems has been traditionally accomplished in compulcr-based process control soft ware by onc or both of the following methods: 1) establishin g limit checks for sensors and raising an alarm whenever a sensor's valuc crosscs one of these thresholds, and 2) comparing predicted values from a simulation to actual sensor values and flagging discrepancies. These anomaly detection techniques arc not as robust as they need lobe. Failures can manifest in ways which arc not captured by these traditional methods. Furthermore, some anomal ous behaviors arc more naturally detected at the level of global interactions affecting multiple sensors,

Wc describe extensions 10 the traditional techniques for anomaly detection, as well as new anomaly detection techniques based on alternate models of what distinguishes normal from abnormal behavior. Some of these techniques arc designed to capture anomalies at individual sensors; some detect anomalies across collections of sensors. To assist in reasoning about complex global behaviors, wc construct and simulate a causal model of the physical system being monitored.

These techniques have been tested on data from the Environmental Control and Life Support System (ECLSS) of Space Station Freedom (SSF) and arc being applied in advanced monitoring prototypes for the SSF External Active Thermal Control System (EATCS) of SSF and the Environmental Emergency and Consumables Management (EECOM) subsystem of the Space Shuttle.

1 Introduct ion

Mission operations personnel at NASA have the task of determining, from moment 10 moment, whether a space platform is exhibiting behavior which is in any way anomalous, which could disrupt the operation of the platform, and in the worst case, could represent a loss of ability to achieve mission goals. A traditional technique for assisting mission operators in space platform health analysis is the establishment of alarm thresholds for sensors, typically indexed by operating mode, which summarize which ranges of sensor values imply the existence of anomalies. Another established technique for anomaly detection is the comparison of predicted values from a simulation to actual values received in telemetry. 1 lowever, experienced mission operators reason about more than alarm threshold crossings and discrepancies between predicted and actual to detect anomalies: (hey may ask whether a sensor is behaving differently than it has in the past, whether a current behavior may lead to a global perturbation or whether a current behavior may lead 10- the particular bane of operators - a rapidly developing alarm sequence.

A fault which propagates through a system faster than the sensor polling rate can create a situation wbcrc, between onc sampling and the next, the number of sensors in alarm goes from zero 10 lens or more. Information about the ordering of events is lost. in ibis kind of emergency situation, operators can experience information over'load and a compromising of their ability 10 interpret the sensor data.

Our approach to introducing automation into real-time systems monitoring is based on two observat ions: 1) mission operators employ multiple methods for recognizing anomalies, and 2) mission operators do not and should not interpret all sensor data all of the time, The subject of this paper is an approach to determining from moment 10 moment which subset of the available sensor data for a system is most informative about the presence of, or potential for, anomalies occurring within the system. Wc term this process *sensor selection* and wc have implemented a prototype selective monitoring system called SELMON [7; 81.

The SELMON system has its origins in a sensor planning system called GRIPE [71 which planned information gathering activities to verify the execution of robot task plans. Other model-based monitoring systems include Dvorak's M IMIC, which performs robust discrepancy detection for continuous dynamic systems [9; 101, and DeCoste's DATMI, which infers system states from incomplete sensor data [51. The SELMON work complements other work within NASA which has focused on empirical and model-base.d methods for fault diagnosis of aerospace platforms [1; 13; 111.

The organizat ion of (his paper is as follows: First we describe sensor importance measures used to identify the presence of anomalies in a monitored system, and 10 appraise the relevance of different sensors for reporting on those anomalies. Two of these measures arc extensions of the traditional techniques of limit sensing and discrepancy detection. The

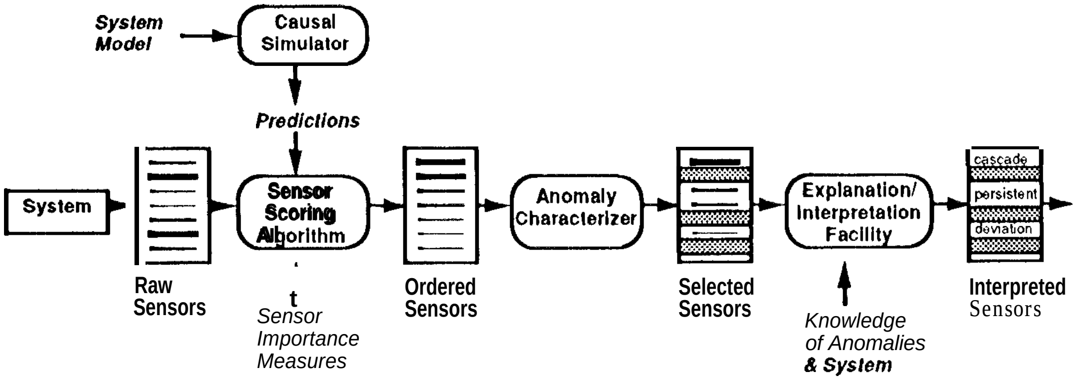

Figure 1: SEI.MON Architecture.

seven measures fall into three categories: those which concern themselves with data provided by single sensors, those which utilize a simulation of the system being monitored, and those which utilize a causal model of the system being monitored. As part of our discussion of the usc of causal modeling and reasoning, wc briefly describe our event-driven causal simulator and our causal modeling language.

Next wc describe the test and application domains wc arc working with, providing mos(detail on the SSF EATCS. Then wc report and discuss results on empirically evaluating the performance of the SELMON system on data from the SS1' ECI .SS testbed. As part of this section, wc describe how SEI .MON detected an anomaly which the traditional monitoring methods faii 10 detect. Finally, wc conclude with some thoughts on future research and applications and a summary.

2 Approach: Selective Monitoring

How dots an intelligent agent – human or machine - observing a complex physical system, decide when something is going wrong? Abnormal behavior is always defined as a departure from normal behavior. Unfortunately, there appears 10 be no single, crisp definition of "normal" behavior. In the traditional monitoring technique of limit sensing, normal behavior is predefined by nominal value ranges for sensors. A fundamental limitation of this approach is the lack of sensitivity to context. A Single fault may manifest in different ways, depending on the configuration Of the system when the fault occurs. The compiled notion of an alarm threshold may not capture these subtleties in manifestation. Another limitation is lilt purely local View of Sensors taken by the limit sensing technique. It is possible for a system to exhibit a pattern of sensor values, each value in its respective nominal range, which is nonetheless inconsistent. Finally, it is never possible to anticipate all fault modes of a system. Since alarm ranges arc predefined during a design analysis in which fault modes of the system arc enumerated, a new fault behavior may noi trigger a predefined alarm at all.

10 the other traditional monitoring technique of discrepancy detection, normal behavior is obtained by simulating amodel of the system being monitored. This approach avoids the limitations of tile limit sensing approach. Normal behavior is fully context-sensitive, being derived Via simulation from currentstate information. And in principle, new faults can be detected because there is no closed-world assumption based on previously and statically enumerated known faults. However, the modci-based approach of discrepancy detection has its own limitations. The approach is only as good as the system model, and some behaviors, e.g., non-linear behaviors, cannot be modeled with high fidelity with existing techniques. In addition, normal system behavior typically changes with time, and the model must Continue 10 evolve. The modeling process must continue both because any long-lived system will degrade, and because a system may be used for different purposes throughout its ii fetime. A good example of the latter is the Voyager 2 mission and spacecraft which were, respectively, replanned and reprogrammed 10 compensate for lower light levels for the extended mission to Uranus and Neptune [121.

Noting the limitations of the existing monitoring techniques, wc have developed an approach to monitoring which is designed 10 make tile anomaly detection process more robust, specifically 10 reduce lilt number of undetected anomalies (false negatives). Towards this cad, wc introduce multiple anomal y models, seven in all , each employing a different notion of "normal" behavior. Two of the seven arc based on the traditional monitoring techniques. For each anomaly model there is a sensor importance measure. These measures determine why, at a particular moment, onc Sensor may be more worthy of operator attention another. The anomaly models and sensor importance measures arc based on concepts from model-based reasoning, statistics, and information theory. While any single one of these sensor importance measures Jnay fail to detect an anomaly, wc will demonstrate how collectively, they provide a finer safety net than the traditional monitoring techniques.

During each timestep all sensors arc scored according 10 these sensor importance measures. The scores arc used as a basis for an ordering on the sensors. See Figure 1,

These sensor scoring measures are partitioned into two broad categories. The first set - empirical methods -- rciy on current and historical data 10 determine importance, and lake a purely local view of sensors, These measures include *surprise*, *alarm*, *anticipate alarm*, and *value change*. The *alarm* measure is based on the traditional notion of limit sensing. Measures from the second set -- model-based methods employ a model of the system and an event-driven simulation capability to reason about current and expected future system performance 10 determine sensor importance. These

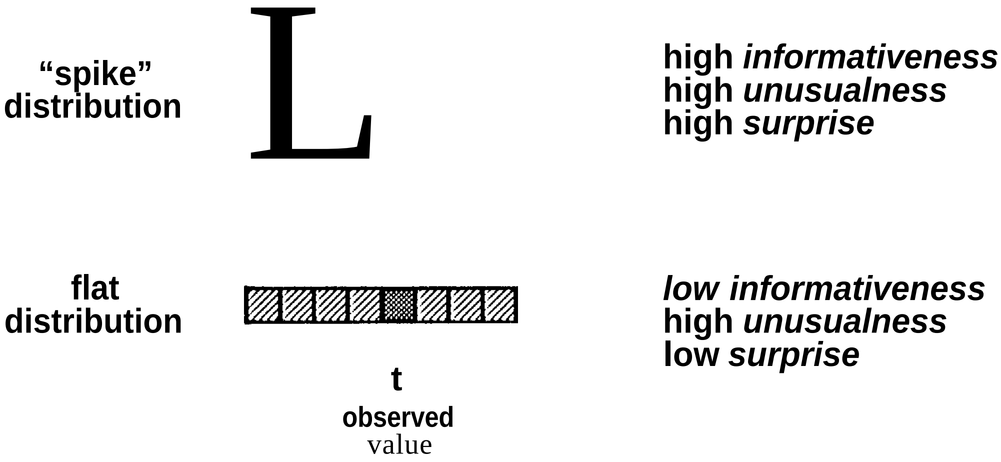

Figure 2: *Surprise* of different sensor values and frequency distributions.

measures include *deviation*, *sensitivity*, and *cascading alarms*. The *deviation* measure is based on the traditional notion of discrepancy detection. The last Iwo measures, *sensitivity* and *cascading alarms*, perform explicit causal reasoning on the model of a system.

This paper will concentrate on our n~odcl-based monitoring methods, providing most detail on our usc of event-driven simulation and causal reasoning techniques. More detail on our empirical monitoring methods can be found in [81.

2.1 Empirical Anomaly Detection Methods

In this section, wc briefly describe the empirical methods that wc usc to determine, from a local viewpoint, when a Sensor is reporting anomalous behavior. There arc four empirical sensor importance measures: *surprise*, *alarm*, *anticipate alarm*, and *value change*. These measures usc knowledge about each individual sensor, without knowledge of any relations among sensors.

Surprise

An appealing way to assess whether current behavior is anomalous or not is via comparison to past behavior. This is the essence of the *surprise* measure. It is designed to highlight a sensor which behaves other than it has historically, Specificall y, *surprise* uscs the historical frequency dist ribut ion for the sensor in two ways: To determine the likelihood of the given current value of the sensor (*unusualness*), and to examine the relative likelihoods of different values of the sensor (*informativeness*). 11 is those sensors which display unlikely values when other values of the sensor arc more likely which gel a high *surprise* score. *Surprise* is not high if the only reason a sensor's value is unlikely is that there arc many possible values for the sensor, all equally unlikely. See Figure 2.

The *informativeness* component of t hc *surprise* measure provides the key 10 detecting a subtle anomal y which is missed by both limit sensing and discrepancy detection, as wilt bc discussed below.

Alarm

Alarm thresholds for sensors, indexed by operating mode, t ypicall yarc established through an off-line anal ysis of system design. The notion of *alarm in* SELMON extends the usual onc bit of information (the sensor is in alarm or it is not), and also reports how much of the alarm rangehas been traversed. Thus a sensor which has gone deep into alarm gels a higher score than onc which has just crossed overthe alarm threshold.

Alarm Anticipation

The *alarm anticipation* measure in SELMON performs a simple form of trend anal ysis 10 decide whether or not a sensor is expected to bc in alarm in the future. A straightforward curve fit is used to project when the sensor will next cross an alarm threshold, in either direction. A high score means the sensor will soon enter alarm or will remain there. A low score means the sensor will remain in the nominal range or emerge from alarm soon.

Value Change

A change in the value of a sensor may bc indicative of an anomaly. In order to better assess such an event, the value *change* measure in SELMON compares a given value change to historical value changes seen on that sensor. The score reported is based on the proport ion of previous value changes which were less than the given value change. It is maximum when the given value change is the greatest value change. seen to date on that sensor. It is minimum when no value change has occurred in that sensor.

2.2 Model-Based Anomaly Detection Methods

Although many anomalies can bc detected by applying anomaly models 10 the behavior reported at individual sensors, the most robust monitoring requires reasoning about interactions occurring in a system and detecting anomalies in behavior reported by several sensors.

The three model-based sensor importance scores in SELMON arc *deviation*, *sensitivity*, *and cascading alarms*. While *deviation* only requires that sonic form of simulation bc available for generating sensor value predictions, *sensitivity* and *cascading alarms* require the ability to simulate and reason with a causal model of the system being monitored.

in the next section, wc describe the usc of cvcnl-driven simulation and explicit causal reasoning 10 support model-based monitoring methods. Towards this cad, wc include

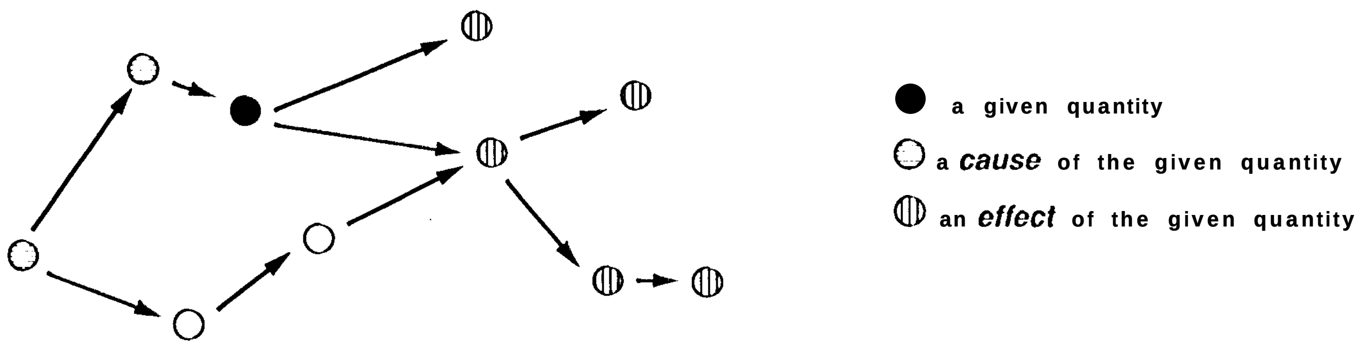

Figure 3: A generic causal graph.

some detail on two tools wc have built within the SELMON project: our EDSE event-driven simulator which accepts a causal model [14], and EDSEL, our causal modeling language [21.

Deviation

The *deviation* measure is our extension of the traditional method of discrepancy detection. As in discrepancy detection, comparisons arc made between predicted and actual sensor values, and differences arc interpreted 10 be indications of anomalies. This raw discrepancy is entered into a normalization process identical 10 that used for the *value change* score, and it is this representation of relative discrepancy which is reported. The *deviation* score for a sensor is minimum if there is no discrepancy and maximum if the discrepancy between predicted and actual is the greatest seen 10 dale on that sensor.

2.2.1 Causal Analysis

Many forms of reasoning relevant to monitoring can be supported by a *causal* model of a system. For the purposes of this discussion, wc define a causal model to be onc which provides explicit information about the quantities of a system, the structural connections through which quantities interact, and the mechanisms which govern how behavior propagates from onc part of the system 10 another.

With this kind of information, it is possible to reason about: How faults may manifest in anomalies al sensors which arc causally downstream and physically distant from the location of a fault. }1ow the information reported at onc sensor may be implicit in snot her, and hence redundant. 1 low several sensors may be in an inconistent state although each is within its nominal range and reporting a predicted value. How an apparently benign current state may contain the seeds of highly undesirable behavior about to manifest in the near future. All of these ideas on how to employ causal reasoning for monitoring purposes arc being explored in the SELMON project. The most fully developed idea at this time is that of using a causal model 10 reason about future behaviors. In SELMON, this reasoning is embodied in the two causal measures *sensitivity* and *cascading alarms*.

Sensitivity measures the potent ial for a large global perturbation 10 develop from current state. *Cascading alarms* measures the potential for an alarm sequence 10 develop from current state. Both of these anomaly measures usc a causal simulator to generate predictions about future states of the system, given current stale. Current stale is taken 10 be defined by both the current values of system parameters (not all of which may be sensed) and the latent events already resident on the simulator agenda. The measures assign scores to individual sensors according to how the system parameter corresponding to a sensor participates in, or influences, the predicted global behavior. Roughly speaking, a sensor will have a high *sensitivity* score when behavior originating at that sensor causes a large amount of change elsewhere in the system. A sensor will have a high *cascading alarms* score when behavior originating al that sensor causes a large number of alarms elsewhere in the system.

Figure 3 shows a generic causal graph. System parameters, some of which will have associated sensors, arc represented by nodes in this graph. Arcs represent the mechanisms which determine if and how system parameters affect other system parameters. I'here arc two node subsets of interest relative to a given system parameter, or quantity node Q. These arc CausesQ, the set of quantity nodes upstream from Q in the directed causal graph, and $Effects_Q$, the set of quantity nodes downstream from Q in the directed causal graph. In computi ng tact ı sen sor's *sensitivity* or *crowding alarms* score, we arc interested only in behavior which "passes through" the quantity corresponding (o the given sensor. The only events processed by the simulator arc those involving the quantity itself, its causes, or its effects. Initially, the simulator only processes events duc 10 the quantity itself or its causes. However, once an event at the quantity itself is processed, the simulator also processes events among the effects of the quantity. These events arc said to derive from the quantity. An event al a quantity which is not reachable from Q is not processed by the simulator, even if it is on the simulator agenda, for such behavior cannot have been influenced by the quantity of interest. The only simulated events which contribute to a sensor's *sensitivity* or *cascading alarms* score arc those involving the quantity itself or its (sensed) effects. Simulation proceeds until a stated future time is reached. Currently, wc assume there is no feedback, i.e., the causal graph is a directed acyclic graph.

Sensitivity

The algorithm used to compute a sensor's *sensitivity* score is as follows:

Given current system state
  (quantity values al t_0 and queued events)
For each quantity Q:
  Retrieve $Causes_Q$
  Retrieve $Effects_Q$
  Simulate only those events originating or deriving from a quantity g $\in$ {Q} $\cup$ $Causes_Q$
  Collect all events at a sensed quantity $Q_e \in Effects_Q$ occurring al lime $t \leq t_0 + \Delta t$:
  Let

$$Sensitivity(Q) = \sum \frac{|Q_e(t) - Q_e(t_0)|}{Max\_Change(Q_e)}$$

t_0 is the current time. At is the maximum time forward 10 simulate. $Max\_Change(Q_e)$ refers to the largest historical value change seen on a given sensor. The values which arc summed arc the ratios between the maximum predicted value change for a given sensor Q_e and the maximum historical value change for that sensor. The *sensitivity* score is maximum when every sensor causally downstream from a given sensor is predicted 10 exhibit its largest value change seen 10 dale. Thus the maximum score for a given sensor is equal to the number of sensors causally downstream from that sensor in the causal graph. The *sensitivity* score is minimum when no changes arc predicted 10 occur causally downstream from a given sensor. All *sensitivity* scores arc zero if the system is in a perfectly steady state.

Note that for a given set of predicted, causally related events, a sensor's score is monotonically higher the closer it is to the causal source of that global behavior. This information may be useful, for example, in aiding operators in selecting a single control action with the greatest impact.

Cascading Alarms

The algorithm used to compute a sensor's *cascading alarms* score is as follows:

Given current system stale
  (quantity values at t_0 and queued events)
For each quantity Q:
  Retrieve $Causes_Q$
  Retrieve $Effects_Q$
  Simulate only those events originating or deriving from a quantity g $\in$ {Q} U $Causes_Q$
  Collect all events at a sensed quantity $Q_e \in Effects_Q$ occurring at time $t \leq t_0 + \Delta t$:
  121

$$CascadingAlarms(Q) = \sum In\_Alarm(Q_e(t))$$

$In\_Alarm(Q_e(t))$ is 1 if the quantity Q_e is in an alarm range at simulated lime t; otherwise it is O. The *cascading alarms* score is maximum when every sensor causally downstream from a given sensor is predicted to be in alarm. Thus the maximum score for a given sensor is equal to the number of sensors causally downstream from that sensor in the causal graph. The *cascading alarms* score is minimum when no alarms arc predicted to occur causally downstream from a given sensor,

Note that the *cascading alarms* score is designed to give monotonically higher scores to sensors that arc closer to the causal source of an alarm sequence. This kind of information may be extremely valuable to an operator attempting 10 interpret a situation where multiple alarms appeared between one polling of the sensors and the next.

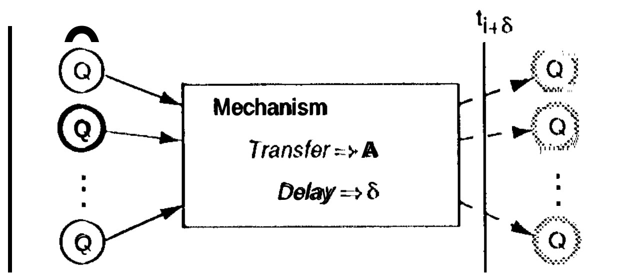

Figure 4: Causal simulation in EDSE: A mechanism evaluated at lime t_i will propagate its value al $t_{i+\delta}$.

2.3 Causal Simulation and Modeling

In this section, wc briefly describe the EDSE cvenl-driven causal simulator and the EDSEL causal modeling language which support the SELMON system.

Causal models arc an abstraction for describing the behavior of a system by representing the physical processes occurring within the system as discrete functions to be evaluated.

A *causal model* characterizes a physical system in terms of state variables and causal influence relations among the variables. In the *causal graph* defined by the state variables and in fluence relations, changes in any variable may be propagated 10 other variables through the influence relations. Causal simulation is the process of tracking changes in Variables and propagating them to other variables through influence relat ions, thereby producing a new set of changes. Implicit in the notion of causality arc the concepts of *event* and *causal time*: Events comprise changes in slate variables duc 10 a specific influence relation and with respect to a specific moment in time. I'hereby, causal lime moves forward duc to delays in the propagation of changes in the causal model.

In our causal modeling language, called EDSEL, causal models arc represented in terms of these primitives. Causal lime is expressed as a monotonically increasing sequence of integer-valued *instants*. Each slate variable is denoted by a *quantity*. Each infl uence relation is described by a *mechanism* which encapsulates a set of input quantities, a set of output quantities, along with *transfer* and *delay* functions. During simulation in EDSE, the transfer function is evaluated with the mechanism input quantities to produce a value that will be propagated to each of the mechanism output quantities. The delay function is simultaneously evaluated 10 produce an offset from the current simulation lime. Value propagation takes place, al the relative time returned by the delay function. Figure 4 visualizes mechanism evaluation. Simulation continues until a user-spccilkd lime limit is reached or the model reaches quiescence.

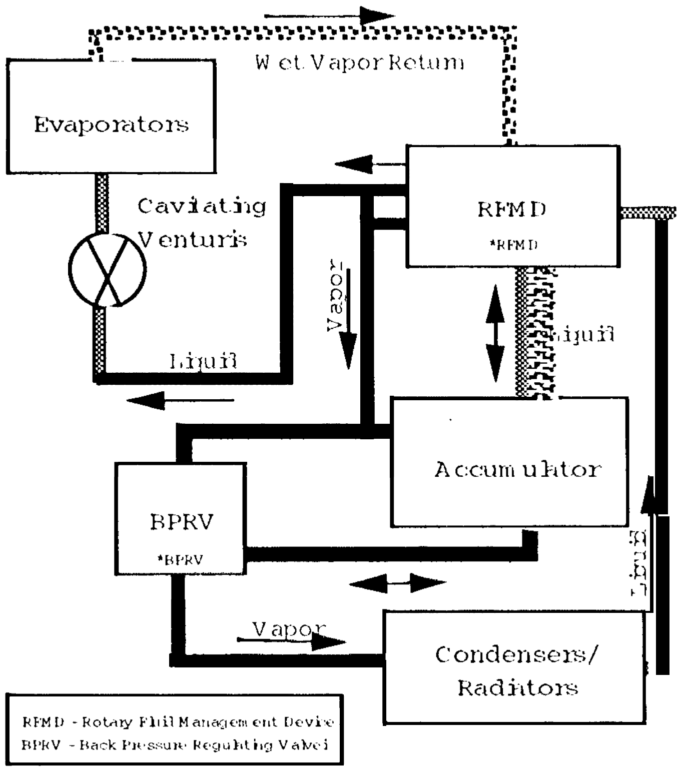

Figure 5: The External Active Thermal Control System (EATCS) of Space Station Freedom.

We adopt a component-centered approach to causal modeling. This modeling approach distinguishes the constituents of the system being modeled as *components* and the structural *connections* among them. A physical component is intuitively defined as a discrete element of a physical system. By analogy, a causal component is an entity that encapsulates the behavior of the corresponding physical component as a whole.

A connection is an abstraction which characterizes the interaction pathways between components. A connection within a physical system enables material or information flow between components. Similarly, causal connection allows quantity values to be propagated between components.

2.3.1 Modeling the SSF EATCS

We have commenced a modeling effort to build a causal model of the SSF EATCS, using our G2-based tool called MESA [3] which generates the EDSEL representation of a causal model from a higher-level abstraction of a component-centered model together with the mechanisms which represent the physical behavior of the components. The MESA-generated model is then loaded into EDSE to perform an event-driven simulation of the EATCS.

Figure 5 represents a simplified schematic of the EATCS, showing all of the major components. The Rotary Fluid Management Device (RFMD) pumps liquid ammonia to the Evaporator loop. Flow control from the RFMD to each of the evaporators is provided by cavitating venturis. In a normal venturi with fixed inlet conditions, reducing the exit pressure results in an increased flowrate. The TCS cavitating venturi passively provide flow stability even with downstream pressure variations resulting from heat load changes. A two-phase fluid mixture of liquid and vapor is generated at each evaporator by the particular heat load being serviced. The two-phase mixture is returned to the RFMD, which separates the liquid from the vapor by centrifugal force. The RFMD pumps the vapor to the radiators, which condense the fluid to a subcooled state by rejecting heat to space. The Back Pressure Regulating Valve (BPRV) maintains setpoint temperature by controlling system pressure. RFMD fluid level changes are due primarily to varying heat loads and are handled by an accumulator with an internal metal bellows separating the liquid and vapor sides.

The component model schematic of a simplified EATCS evaporator loop is shown in Figure 6. Two cavitating venturis, CV1 and CV2, regulate flow to two evaporators, EV1 and EV2, operating in parallel. The RFMD is modeled as a flow and pressure source for the evaporator loop, and divide and join tees DV1 and JT1 represent the flow junctions in the loop. H1 and H2 represent the heat sources for the evaporators. The modeler creates instances of these components from a component library which contains the definitions of component classes, their input and output quantities, and the mechanisms which encapsulate the transfer and delay functions with the quantities. An example of a component definition for the cavitating venturi is shown in Figure 7. This representation is built by a developer in MESA for the component library, and the instantiation of the actual underlying component quantities, transfer and delay functions, and the inter-component connections is handled automatically by MESA when the component model schematic is built. Thus, the modeler is freed from details of building the causal model directly in EDSEL.

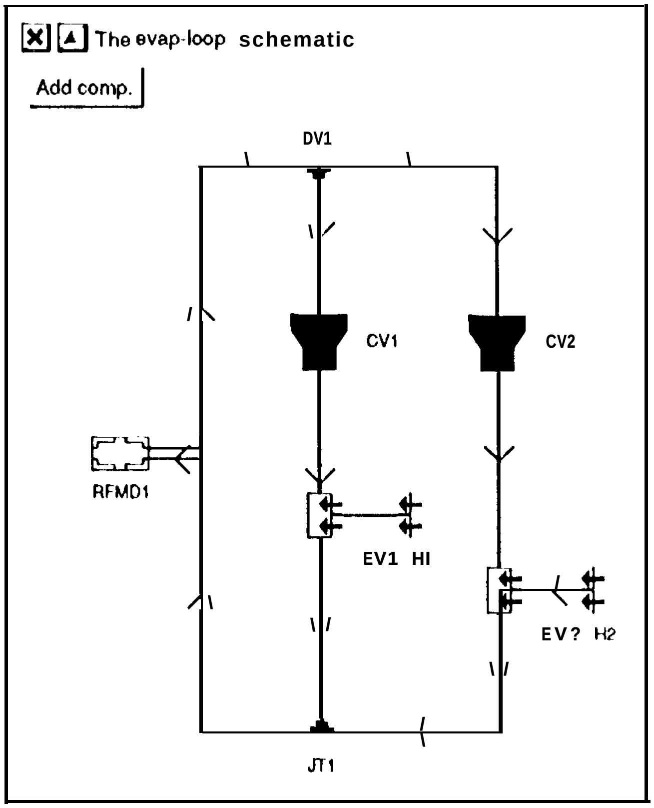

Figure 6: The Evaporator Loop of SSF EATCS.

The component definition of the venturi as shown in Figure 7 shows the flow of input quantities through transfer functions to output quantities for each state variable that describes a cavitating venturi. These transfer functions arc based on the laws of physics, fluid dynamics and thermodynamics, and contain a complete mathematical description of the steady slate behavior of the device. Each transfer function by definition may have multiple inputs, but produces a single output. In cases where intermediate calculations must be made to determine an output quantitiy, such as for saturation pressure and liquid density at the venturi inlet temperature used for flow rate determination, mechanisms arc chained together to produce the final result.

Using these generic modeling techniques, a causal model of the simplified EATCS evaporator loop has been generated in EDSEL, and simulations were executed successfully in EDSE. Wc are currently working to model the EATCS radiator loop and the remainder of the RFMD and BPRV. The next step will be to incorporate sensor objects into the model to support the model-based sensor importance measures in SELMON.

3 Performance Evaluation

The SELMON sensor importance measures are dynamically computed each time the sensors arc polled. In order to assess whether SELMON usefully focuses operator attention while performing robust anomaly detect ion, we performed the following experiment: Wc evaluated SELMON performance in selecting critical sensor subsets specified by an SSF ECLSS domain expert, sensors seen by that expert as useful in understanding episodes of anomalous behavior in actual historical data from ECLSS testbed operations.

The experiment asked the following two specific questions: How often did SELMON place a "critical" sensor in the top half of an overall ordering on the sensors'? and How does SELMON performance compare to traditional monitoring practice?

Wc used two alternate ways of combining the scores from the seven sensor importance measures to arrive at an overall ordering: 1) the individual scores were composed arithmetically (see [81 for details) and 2) the maximum individual score was taken. Wc also ran SELMON with only the *alarm* measure active, to represent traditional monitoring.

The performance of a random sensor selection algorithm would be expected to be about 50.070: any particular sensor would appear in the top half of the sensor ordering about half the time. Table I shows the results of our experiment. The first column identifies onc of the anomaly episodes specified by the domain expert, The second column shows the overall "hit" rate for each episode using the *alarm* measure only, i.e., the percentage of lime SELMON placed the given sensor in the top half of the sensor ordering generated by the *alarm* measure. The third column shows the hit rate using arithmetic composition of the scores. The fourth column shows the hit rate when the maximum score is used. Finally, the fifth column shows the hit rate when either way of combining the scores is used.

EPISODE	Alarm	Composed	Max	Either
High Flow Rate	97.8	94.5	83.5	94.s
Sensor Malfunction	100.0	100.0	100.0	100.0
Unibed Loading	48.9	56.6	90.6	90.6
Pre-Healer Off	100.0	100.0	81.1	100.0
Emergency Shutdown	100.0	100.0	100.0	100.0
Pressure Fluctuations	98.6	100.0	90.1	98.6
High Pressure	92.5	95.5	91.0	92,5
All	76.3	79.8	88.1	9s.1

Table 1: SELMON Performance at selecting critical sensor data.

These results show that SELMON performs much better than random at replicating the at tent ion focusing of an ECLSS domain expert. More to the point, when the maximum sensor importance measure is used for each sensor, SELMON performs considerably better than the traditional practice of limit sensing, Finally, when both methods of combining the individual scores are available (noting that, at the current time, wc do nol have an automated technique for determining in context which method is more appropriate, and such determination would be left to the operator), SELMON performance is quite notable, detecting actual anomalies 95% of the time.

These results lend credibility to our premise that the most effective monitoring system is one which incorporates several models of anomalous behavior, and takes multiple views of sensor importance. The best, most experienced mission operators arc already remarkably effective at knowing when something is going wrong on a space platform, Our aim is to offer a more complete, more robust set of techniques for anomaly detection, to make mission operators even more effective, or to provide the basis for an automated monitoring capability.

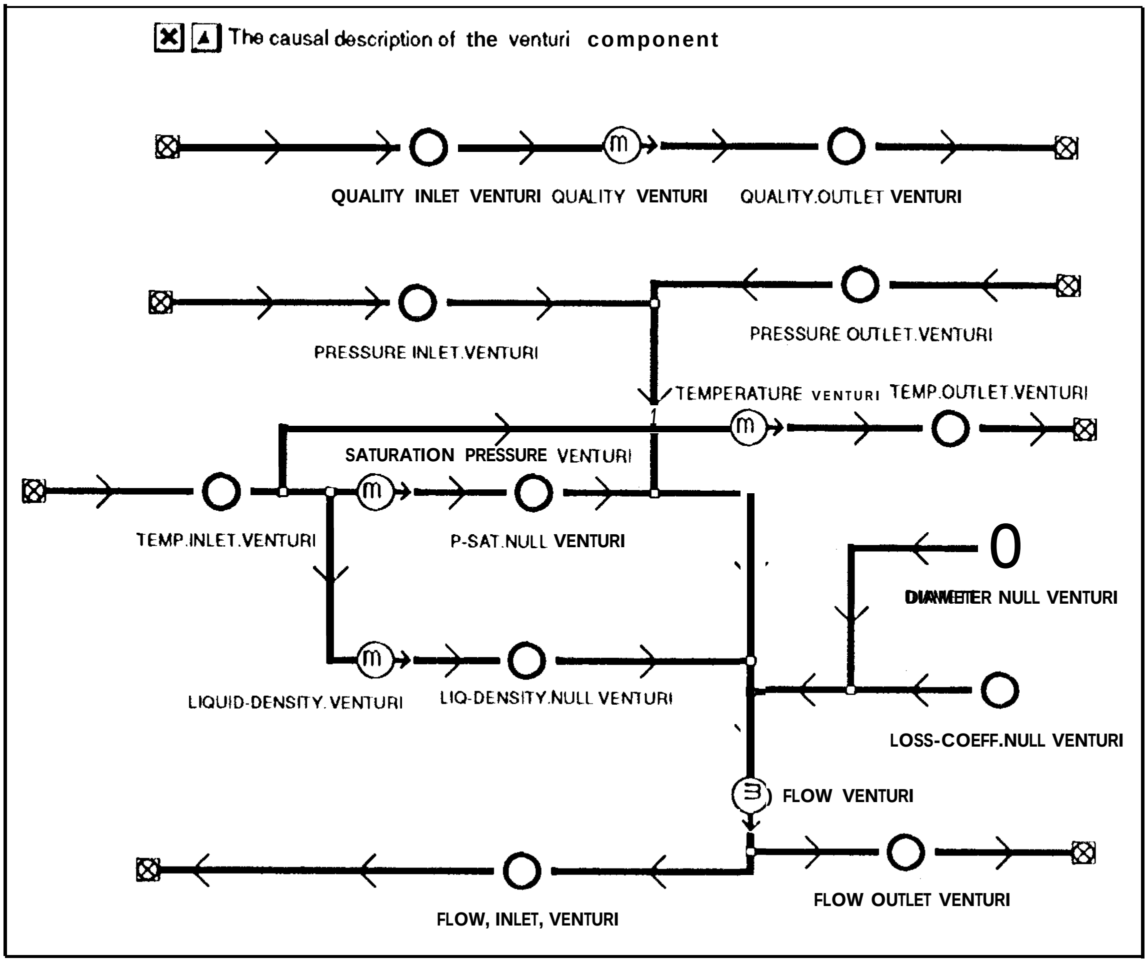

Figure 7: The Venturi component of SSF EATCS.

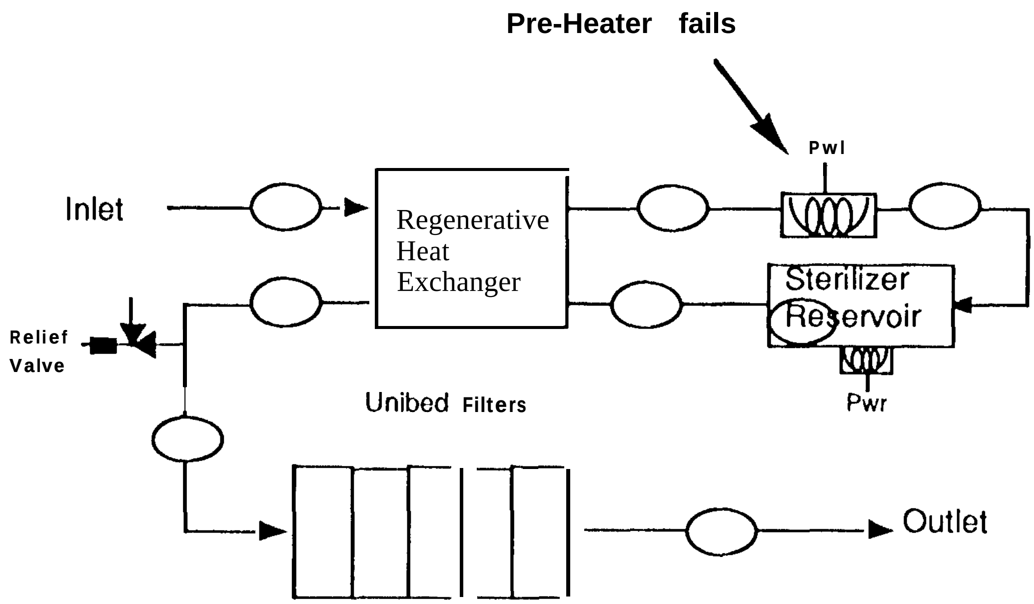

Figure 8: A failure on the waler side of SSF ECLSS.

We plan 10 extend this empirical analysis of SELMON performance in two directions: 1) using both the *alarm* and deviation measures as the baseline for performance comparison, to better represent traditional monitoring techniques, and 2) 10 examine the performance of the causal measures sensitivity and *cascading alarms* separately, possibly in artificial domains, where the topology of the causal graph and the mechanisms can be specified to better analyze how these measures perform under different situations of undesirable behavior propagation.

Besides examining SELMON performance at the statistical level, we have also examined individual examples of successful anomaly detection in SELMON. The following example illustrates how SELMON highlights a subtle manifestation of an anomaly which the traditional monitoring techniques fail to detect.

During an episode when the ECLSS testbed pre-heater failed (see the schematic in Figure 8), system pressure, which normally oscillates within a known range, became more stable (see Figure 9), This "abnormally normal" behavior is not detected by limit sensing because the system pressure remains firmly in the nominal range; nor by discrepancy detection, for the predicted value is roughly the average of the values over time, this oscillating behavior being unmodelled. However, the SELMON *informativeness* component of the *surprise* meas urc ri scs during this episode. *Informati veness* rises when the frequency distribution across the range of sensor values moves away from a flat distribution towards a "spike" distribution (see Figure 2). A suddenly stable system pressure results in one of the value ranges for system pressure beginning to dominate the frequency distribution (see Figure 10). SELMON provides the means of detecting and reasoning about this kind of subtle anomaly.

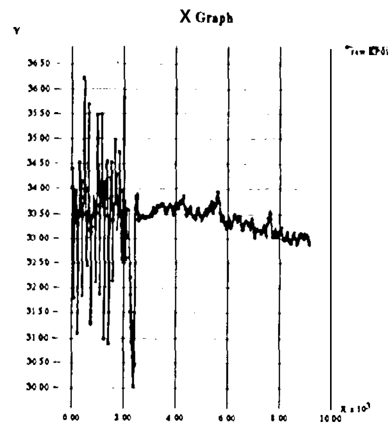

Figure 9: **ECUS System Pressure During Abnormally Normal Episode.**

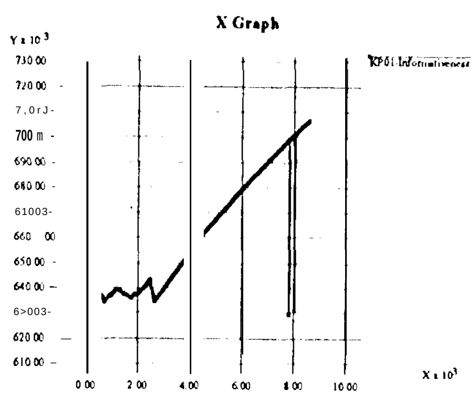

Figure 10: *Informativeness* **of ECLSS S ystem Pressure During Abnormally Normal Episode.**

4 Towards Applications

Wc arc working to apply the SELMON techniques in an advanced monitoring and diagnosis prototype for the SSF External Active Thermal Control System (EATCS). The Thermal Control System Automation Project (TCSAP) [11] is developing a Jmow]cdgc-based system to perform Fault Detection, isolation and Recovery (FDIR) on the EATCS. The TCSAP software is a hybrid advanced automation system implemented in a commercial rczzl-lime expert system shell called G2. 11 uses a combination of conventional programming, rule-based technology, and mode]- based reasoning 10 provide extensive diagnostic power and flexibility beyond the capabilities based on alarm thresholds and limit sensing. Within TCSAP, the SELMON empirical sensor importance measures described earlier have been implemented, and testing and evaluation have begun in ground-bawd Thermal Test Bed runs at NASA Johnson Space Center (JSC).

Also at NASA JSC, wc arc exploring applications of the SELMON approach to Space Shuttle (STS) operations. In particular, wc arc examining the Environmental Emergency and Consumables Management (EECOM) subsystem of STS. This is the life support system for STS, the analog of ECLSS for SSF.

The seven anomaly measures in SELMON arc designed 10 be applied separately. This feature allows a SELMON-based application to be prototyped and evaluated in an efficient manner. As long as sensor data is available, on-line or historical, the four empirical (non-rnodcl-based) anomaly measures can be immediately tested. If any form of simulation is available, the *deviation* measure also can be tested. It is only the two causal measures which must await for the causal modeling process to complete. Thus SELMON avoids a common limitation of Al-based systems development: that no delivery is possible, even a prototype system, before a large knowledge engineering or modeling investment is made.

5 Future Work

An unresolved area in SELMON is developing a wc]i-founded method for utilizing all the sensor importance measures, Wc have working concepts for how 10 compose the individual measures into a total sensor importance score, but wc consider this an area for further work, A theoretical or empirical analysis may provide insight on the most appropriate composition technique, onc that is tailorable to different applications.

Wc recognize that an important component of the SELMON approach is the ability to provide explanations or interpretations of why a particular sensor has been highlighted and is more worthy of operator attention than other sensors. Other future work in the SELMON project wiii complement existing sensor ordering and anomaly detection capabilities with modc,l-based capabilities for characterizing anomalies by their temporal and spatial extent, and focusing attention according 10 multiple viewpoints (causal priority, proximity 10 control points, potential for irreversible damage, tic.).

la related work, wc arc also investigating the problem of sensor placement during design [4].

6 Summary

Wc arc developing techniques 10 support real-time monitoring through sensor selection, the moment to moment focusing of attention on a subset of the available sensor data. Sensor selection is based on a set of importance criteria using different models of what constitutes an anomaly. The computational realizations of these importance criteria draw on concepts from model-based reasoning, statistics, and information theory. Experimental results show that our sensor selection techniques arc cffcc.live at highlighting the sensors deemed critical by a domain expert for understanding actual anomalous episodes from the Space Station Fredom Environmental Control and Life Support System testbed. These results also suggest that a monitoring system which employs multiple models of anomalous behavior is more effective than onc based on the traditional monitoring concepts of alarm thresholds and discrepancies.

7 Acknowledgements

Others who have worked recentl y on the SELMON project include Steve Chien, Daniel Clancy, Usama Fayyad and Harry Porta.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California last it ute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] K. }1. Abbott, "Robust Operative Diagnosis as Problem Solving in a Hypothesis Space," *7th National Conference on Artificial Intelligence*, St. Paul, Minnesota, August 1988.

[2] L. Charest, Jr. and R. Faltisco, "Building Causal Models for SELMON: A User's Manual for the EDSEL Modeling Language," JPL Publication, *in preparation*.

[3] L. Charest, Jr., N. Rouquette, R. Doyle, C. Robertson, and J. Wyatt, "MESA: An Interactive Modeling and Simulation Environment for Intelligent Systems Automation," submitted 10 *Hawaii International Conference on System Sciences*, Maui, Hawaii, January 1994.

[4] S. A Chien, R. J. Doyle, and N. F. Rouquette, "A Model-based Reasoning Approach to Sensor Placement for Diagnosabil it y," *2nd International Workshop on the Principles of Diagnosis*, Milan, Ital y, October 1991.

[5] D. DeCoste, "Dynamic Across-Time Measurement Interpretation," *8th National Conference on Artificial Intelligence*, Boston, Massachusetts, August 1990.

[6] R. J. Doyle, I). J. Atkinson, and R. S. Doshi, "Generating Perception Requests and Expectations to Verify lilt Execution of Plans," *5th National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, August 1986.

[7] R. J. Doyle, D. Berleant, L. Falcone, and U. Fayyad, "Selective Simulation and Selective Sensor Interpretation in Monitoring," *Computers in Aerospace S*, American Institute of Aeronautics and Astronautics, October 1989.

[8] R. J. Doyle, S. A. Chien, U. M. Fayyad, and E. J, Wyatt, "Focused Real-time Systems Monitoring Based on Multiple Anomaly Model s," *7th International Workshop*

on Qualitative Reasoning, Eastsound, Washington, May 1993.

[9] D. L. Dvorak and B. J. Kuipers, "Model-based Monitoring of Dynamic Systems," *11th International Conference on Artificial Intelligence*, Detroit, Michigan, August 1989.

[10] D. L. Dvorak, "Monitoring and Diagnosis of Continuous Dynamic Systems using Semiquantitative Simulation," Ph.D. dissertation, The University of Texas at Austin, May 1992,

[11] T. Hill, W. Morris, and C. Robertson, "Implementing a Real-time Reasoning System for Robust Diagnosis," *6th Annual Workshop on Space Operations Applications and Research*, August 1992.

[12] R. P. Laeser, W. I. McLaughlin, and D. M. Wolff, "Engineering Voyager 2's Encounter with Uranus," *Scientific American*, 255(5):36-45, November 1986.

[13] J. Muratore, T. Heindel, T. Murphy, A. Rasmussen, and R. McFarland, "Space Shuttle Telemetry Monitoring by Expert Systems in Mission Control," in *Innovative Applications of Artificial Intelligence*, H. Schorr and A. Rappaport (eds.), AAAI Press, 1989.

[14] N. F. Rouquette, S. A. Chien, and L. K. Charest, Jr., "Event-driven Simulation in SELMON: An Overview of EDSE," JPL Publication 92-23, August 1992.

[15] E. A. Scarl, J. R. Jamieson, and E. New, "Deriving Fault Location and Control from a Functional Model," *3rd IEEE Symposium on Intelligent Control*, Arlington, Virginia, 1988.